leaving the value of date property as previously set if the date property is marked as a hard date; and

executing each action component associated with the action date component based on the point in time specified in the date property of the action date component.

COMMENTS

This Amendment B is responsive to the first, non-final, Office Action mailed December 19, 2003 (the "Office Action"). Applicant requests a one month extension for response. A check is enclosed with this Amendment B for the required $55 fee pursuant to 37 CFR 1.17(a)(1).

Claims 2-21 are pending in this Application and all stand rejected under the Office Action. Applicant respectfully traverses the claim rejections specified in the Office Action and requests reconsideration or further examination of the application pursuant to 37 CFR 1.111. Applicant further requests entry of this Amendment B pursuant to 37 CFR 1.112.

All pending claims have been rejected as anticipated under 35 USC 102(b) by Matoba et. al., U.S. patent number 5,479,343, issued on December 26, 1995 (hereinafter "Matoba"). As the examiner is no doubt aware, a proper rejection of a claim as anticipated under 35 USC 102(b) requires that "each and every element as set forth in the claim is found, either expressly or inherently described in a single prior art reference. The identical invention must be shown in as complete detail as is contained in the ... claim." [Citations omitted]. MPEP §2131 (August 2001). The following comments will demonstrate that Matoba not only does not teach each an every element of the rejected claims, it fails to describe the present invention in any substantive way. Before addressing the claims in detail, a brief, high-level, description contrasting Matoba and the present invention is provided.

Mark I. Crohn
Reg. No. 40573
109 2nd St. S., #429
Kirkland, WA 98033
(425) 828-7710

A Brief Review Of Matoba.

Matoba teaches a fairly complex production planning system which, with the assistance of considerable human operator intervention, helps plan the production of unspecified items. Applicant is unable to find any indication that the system actually produces anything, but is rather a sophisticated scheduling guide that employs many specialized devices to adjust a proposed schedule based on the analysis of a number of external factors such as shop availability, empirically determined "lateness" of various production steps, prequalification of the production steps, interactive adjustments made by a user using the system, etc. These devices feed their conclusions to a central "MRP Calculation Control Device," which adjusts the centralized, proposed schedule accordingly. There are numerous graphical displays that employ conventional workflow diagrams that show contiguous blocks, each representing a block of time used for that production step.

At its core, Matoba teaches a sophisticated production scheduling aid that requires considerably human interaction to operate. It displays information using conventional workflow-type diagrams and provides conventional interface controls that allow for the human operator to input/adjust the values. The calculation of the effect of adjustment values is the most the complicated area of the Matoba invention. In particular, Matoba employs techniques to integrate several production steps and production chains (e.g., shops) into a single production timeline.

Undoubtedly, the system described in Matoba took many hundreds of person hours to construct using entirely conventional programming techniques. The Matoba program is dedicated to a specific use. While providing many tools to assist a human operator with production scheduling, its disclosure suggests no adaptability to other uses beyond production scheduling or extensibility beyond what it is specifically programmed to do.

Mark I. Crohn
Reg. No. 40573
109 2nd St. S., #429
Kirkland, WA 98033
(425) 828-7710

A Brief Review Of The Present Invention.

In contrast to the invention of Matoba, the present invention is not a program dedicated to a specific and restricted use as a production planning system. Rather, the present invention provides a method and system for programming and executing groupings of time sequenced tasks in almost unlimited permutations for almost unlimited applications. The present invention greatly simplifies the task of programming and executing groups of time sequenced tasks, putting not only use of the resulting program within the grasp of the average user, but also enabling an average user to actually program, reuse, combine, expand, or adapt the groups of time sequenced tasks. This ability is enabled by a provided novel object model, which not only allows for almost infinitely extensible and reusable automated task groupings, but also allows for reuse by automatically discovering and setting the times and information necessary for the tasks based on the association between provided components and preconfigured time offsets associated with some of those components.

The following brief example illustrates one of many possible uses for the present invention. There are also references to particular claims for the Examiner's use in understanding the claims in the context of this particular example. The references are included for the Examiner's convenience in navigating the claims and are not intended to suggest that the indicated claims may only be interpreted in the context of the example described below, nor is it suggested that there are not other claims that apply. The specifics of the implementation of the example should also not be read into the claim as additional limitations. Since use of the invention often employs multiple instances of each of the three basic components provided by the described embodiment of the invention, attention must be paid to the reference number of each object in order to properly understand the example. Of course, the following discussion is intended to be

Mark I. Crohn
Reg. No. 40573
109 2nd St. S., #429
Kirkland, WA 98033
(425) 828-7710

illustrative of one truncated example only and the breadth of the invention and the scope of the claims should be determined from the full disclosure and claims, respectively.

Example A, attached to the end of this Amendment B, graphically shows a Bolt constructed in accordance with the present invention. The purpose of this Bolt is to respond to a non-final office action, such as the one that you are reading. Example B, also attached to the end of this Amendment B, shows parts of the same Bolt in an alternate format as a checklist. Once constructed, the Bolt can be used for any or all responses to non-final office actions (as is described in the specification with reference to the reuse of bolts and sub-bolts.) In this example the components are implemented using objects and object-oriented programming techniques.

The embodiment of the invention used in this example provides three basic objects: action date objects, action list objects and action objects. These objects are illustrated in Example A by action date graphic representations 2, 4, 6 and 8; action list graphic representations 10, 12, 14, 16, and 18a-b; and action graphic representations 22, 24, 26a-b, 28a-b, 30a-b and 32a-b. The construction/programming of the Bolt comprises generally the subject matter of Claims 8-16. For example, the action date object and the actions object recited in independent claim 8 are represented in this example by the action date graphic representations 2, 4, 6 and 8 and action graphic representations 22, 24, 26a-b, 28a-b, 30a-b and 32a-b. The action list object introduced in dependant claim 11 is represented in this example by the action list graphic representations 10, 12, 14, 16, and 18a-b. For convenience in this description, the objects and their graphic representation will collectively be referred to as objects, even though the objects provided by the present invention may be used with or without graphic representations (e.g., through an API). The graphic interface is recited beginning with dependant claims 9 and 12.

The Bolt is constructed by generating instances of the action date, action list and action objects, as desired, and associating them together to form a pattern with a

Mark I. Crohn
Reg. No. 40573
109 2nd St. S., #429
Kirkland, WA 98033
(425) 828-7710

particular functional meaning. Using the graphic interface this may be done by graphically connecting the object using the indicated lines, drag-dropping the objects on each other, manually setting connections, etc. This association will be referred to as "adding" the object to another object. The associations between the objects that form the pattern determine certain properties of the individual objects and how the Bolt will execute (described below).

Claims 2-7 read on the resulting Bolt and Claims 17 reads on the execution of the Bolt. Usually the graphic user interface will provide by default a representation of a start object 2, which is a sub-classed form of an action date object. The start object is more fully discussed in the specification, but for the purposes of this example, acts as the starting point for the Bolt. For this example, a date property associated with this start object is set to December 19, 2003 – the mailing date of the Office Action. (See, e.g., Claim 2). An action list object 10 is added to the start object 2 and named "Non-final Office Action Response." (See, e.g., Claim 4). This action list object 10 and its progeny comprise an automated task list for responding to non-final office actions and may be reused at will as discussed below.

As described in the specification, action objects are generally sub-classed and provided with execute methods that actually perform a wide variety of configurable tasks. This example assumes using a variety of these sub-classed action objects. The first action that will be taken when responding to the office action is indicated by an action object 22. This action object 22 has an associated task that sets docket dates in an external docket system associated with the office action response. (See, e.g., Claim 2).

The next series of tasks take place at a date associated with another instance of the action date object 4. (See, e.g., Claim 3). This second instance of the action date object 4 is added to the action list object 10, which, in effect, passes thru this association to the first instance of the action date object 2. (See, e.g., Claim 4). An offset value associated

Mark I. Crohn
Reg. No. 40573
109 2nd St. S., #429
Kirkland, WA 98033
(425) 828-7710

with the second action date object 4 is given a value of three months. As will be discussed further below, the date associated with the second action date object 4 is dynamically set based on the association of the second action date object 4 with the first action date object 2 (via the action list object 10) and the offset value associated with the second action date object 4. (See, e.g., Claim 3). In other words, the date associated with the second action date object will be dynamically set to December 19, 2003 plus 3 months, or March 19, 2003 (corresponding to the end of the shortened statutory period).

Unfortunately, we all know that the shortened statutory response period may or may not be met, and like this response, may take advantage of one or more extensions. In this example, the aspect of the present invention that allows for the association of action lists with contexts provides for this circumstance (See, e.g., Claim 7). Regardless of whether the shortened response period deadline is met, certain actions should take place. This is accomplished by adding an "Always" action list 12 to the "Always" context of the second action date object 4. To provide for a reminder that occurs 15 days before the response is due, a third action date 6 is added to the "Always" action list and given an associated offset value of -15 days. According to a method provided by the invention, a third date value associated with the third action date object 6 is automatically set by the present invention by discovering the date value associated with the second action date 4 (discussed below) and adding/subtracting the offset value associated with the third action date object 6. A third action list 14 is added to the third action date object 6 and an action 24 is added to the third action list 14. The third action 24 is configured to send an email reminder to the responsible attorney on the date associated with the third action date object.

The date value associated with the second action date object 4 is discovered by virtue of the association of the third action date object 6 with the second action date object 4 (through the action list object 12). Once the present invention determines the

Mark I. Crohn
Reg. No. 40573
109 2nd St. S., #429
Kirkland, WA 98033
(425) 828-7710

association between the third action date object 6 and the second action date object 4, the offset value associated with the third action date object 6 is added (negative value subtracted) from the date value associated with the second action date object 4 to set the date associated with the third action date object. In this example the date associated with the second action date object 4 is March 19, 2004 from which the offset value associated with the third action date object (-15 days) is subtracted to set the date associated with the third action date object 6 equal to March 4, 2004.

Note that to this point in the example that the dates associated with both the second action date object 4 and third action date object 6 have now been set automatically by a method provided by the invention through their associations with each other and with the start object 2 and then calculated using their own offset values. Note also that setting the date associated with the start object 2 to a different value would propagate the setting of a new date value to each of the second action date object 4 and the third action date object 6. This behavior is one aspect of the present invention which makes the entire bolt reusable for another office action response simply by instancing a new copy of it and either setting a new date in the start object (or attaching it to another bolt and discovering a new date from a parent action date object).

The second context demonstrated in this example is a "True" context of action date object 4. In this example, an "Office Action Documents" action list object 18a is added to the "True" context of the action date object 4. Four actions objects 26a, 28a, 30a and 32a are added to this action list object 18a. The present invention provides the ability for the individual action objects to discover matter information (matter numbers, database connections, SQL strings, etc.) through that action object's association with the Bolt it is associated with (See specification for details).

Each of the action objects attached to the "Office Action Documents" action list object 18a uses the discovered matter information to fill-in fields in template documents.

Mark I. Crohn
Reg. No. 40573
109 2nd St. S., #429
Kirkland, WA 98033
(425) 828-7710

(See, e.g., Claim 18). Action object 26a generates a transmittal letter to be submitted with the office action response, action object 28a finalizes and prints office action response and action object 30a generates and prints the receipt postcard submitted with the office action response. To reinforce that action objects perform many different types of tasks, the action object 32a uses the matter information to make a billing entry in a billing system indicating the that office action response has been generated. (See, e.g., Claim 17).

A third context provided in this example is the "False" context. The False context executes when the methods of the present invention determine that certain conditions have not been met and the True context should not execute. (See, e.g., Claim 19). A false context action list 16 is added to the action date object's 4 false context. If the false context is executed by invention, the actions that are to occur will do so at a later time. This later time is implemented by adding another instance of an action date object 8 to the false context action list 16. The offset value associated with the action date object 8 is configured to 1 month. Looking back via its association to the action date object 4, which has an associated date value of March 19, 2004, the invention automatically adds the offset value of one month associated with the action date object 8 to set that objects associated date to April 19, 2004.

The actions that take place on the one month extension date (April 19, 2004) are identical to what would have been done if the original 3 month deadline had been met. Therefore, the addition of another instance of the "Office Action Documents" action list object 18b is added to the True context of the action date 8. (See, e.g., Claim 15). This second instance of the Office Action Documents action list 18b is comprised of action objects 26b, 28b, 30b and 32b, which correspond to items 26a, 28a, 30a and 32a of the first instance 18a. (See, e.g., Claim 16). The use of multiple instances of sub-bolts demonstrates another useful and novel aspect of the present invention.

Mark I. Crohn
Reg. No. 40573
109 2nd St. S., #429
Kirkland, WA 98033
(425) 828-7710

Once assembled as discussed above, the Bolt 1 may be executed independently, added to another bolt or copies of the Bolt 1 or instanced and added to any number of other bolts. Assuming for this example that the Bolt 1 is added to a matter corresponding to this patent application. If necessary, the methods provided by the present invention will update the action date objects as discussed above. Recall that it has been described above that the action date objects have been set as follows: start object 2 (December 19, 2003; the mailing date); action date object 4 (March 19, 2004; the 3 month date); action date 6 (March 4, 2004; a reminder date) and action date object 8 (April 19, 2004; the one month extension date). (See, e.g., Claim 17). Since the action object 24 is attached to the "Always" context of the action date object 6, the associated task will cause an email to be sent to the responsible attorney on March 4, 2004. Given that the office action response was not ready by the 3 month date, the methods of the present invention would execute the False context of the action date object 4. This would send the execution of the Bolt 1 to the action date object 8, which represents the one-month extension date. (See, e.g., Claim 19).

Since the response is now ready, the "True" context of the action date object 4 would be executed by the present invention, causing the action objects 26b, 28b, 30b and 32b to execute their associated tasks. These associated tasks generate and print the documents submitted with this Amendment B and make a billing entry in the firm billing system associated with these tasks.

Now assembled and configured, the Bolt may be saved as an automated task list and may be reused at will by attaching the Bolt 1 to other matters or other bolts. (See, e.g., Claim 8). For instance, if another non-final office action was received in this or another matter, another instance of the Bolt could be added to that matter. By setting the date associated with the start object 2 in that new instance to the mailing date of the new

Mark I. Crohn
Reg. No. 40573
109 2nd St. S., #429
Kirkland, WA 98033
(425) 828-7710

non-final office action, all the following action dates would be updated to reflect the due dates applicable to that new office action.

CLAIMS ARE DISTINCT

From the preceding discussion, it is apparent that Motoba does not teach or suggest at least each of the following aspects of the present invention:

1. Setting points in time associated with discrete components based upon their association with other discrete components.

2. Executing task methods associated with action components at or near points in time associated with action date components.

3. Executing a task method associated with an action component based upon a context of an action date component with which it is associated.

4. Providing a graphical interface to associate components with each other in functional patterns that permit them to discover and set their associated properties based on their association with other components.

5. Providing extensible and reusable components in either a programming method or system that uses such components.

The preceding represents only a small portion of the substantial differences between Matoba and the present invention. As mentioned above, a single omission in Matoba means that the present invention has not been anticipated. MPEP §2131 (August 2001). Particular distinctions are addressed below in a detailed analysis of the claims.

Independent Claim 2

Claim 2 recites two components that find no corresponding structure in Matoba:

> "an action date component having an associated date property that specifies a point in time" and
>
> "an action component having an associated task method"

Mark I. Crohn
Reg. No. 40573
109 2nd St. S., #429
Kirkland, WA 98033
(425) 828-7710

The term "component" has a recognized meaning in the art of computer applications and systems. A "component" is defined by one source as "1. A discrete part of a larger system or structure. 2. An individual modular software routine that has been compiled and dynamically linked, and is ready to use with other components or programs." *Microsoft Computer Dictionary* (4th Edition 1999). Matoba does not describe the recited components according to either definition or common understanding in the art, particularly when the components are recited as having an associated date property and an associated task method, respectively.

Matoba does not teach an association between components in general and between an action date and action component in particular. Claim 2 has been amended to explicitly recite the industry understanding that associating a component with another component includes making the component aware of the other component such that the component can access certain of the properties, methods or events of the other component. In some computer languages this is implemented by providing a reference or a pointer to the other component that contains a memory address of that component. Associating a property or method with a component includes making the data or process associated with that property, event or method available to that component. Matoba merely groups data items together in tables, which is understood not to be an association between components.

Claim 2 has also been amended to clarify that a task associated with action component is an executable method. As recited in this claim, a task method associated with the action component is executed based on the date property of an action date component with which it is associated. As discussed above, the association between instances of these components and the segregation of the date property in one component and the task method in another component provides a mechanism for almost unlimited flexibility or extensibility that is not taught or suggested by Matoba.

Mark I. Crohn
Reg. No. 40573
109 2nd St. S., #429
Kirkland, WA 98033
(425) 828-7710

The Office Action cites Matoba col. 18 lines 31-64[1] as teaching elements of this claim. This portion of Matoba describes the display of a production chain in a production chain display part 53. The production chain is displayed in a "predefined format" that resembles a conventional workflow diagram, i.e., the production chain moves from shop to shop (production step to production step) as the part moves through its production, with graphic bars representing the time of production. The production chain is generated by a centralized problem area analysis device 7. The underlying data supporting the graphic display elements can be manually changed by a human operator using dials 59-61, and the changes processed by the centralized MRP device 2. Therefore, among many differences with the present invention, the production chain does not represent the association of components as recited in the present claims.

The Office Action also cites Matoba col. 20 line 33- col. 21 line 55[2]. This portion of Matoba teaches a graphic user interface (GUI) through which a human operator adjusts a production planning system. The GUI provides a number of screen elements that the human operator uses to adjust the production schedule. For example, the GUI provides a lateness/margin bar graph 79 that graphically displays the difference between the PRESENT and STARTING date in the black portion of the graph and the "margin" available that is available for a "lateness" compensation between the COMPLETION and DELIVERY DATE in the white portion of the graph. The human operator can manually adjust the COMPLETION date by activating the completion date change dial 74. This manual process can be repeated for different orders by activating the order number change dial 73. Once the manual change is made, the result is sent to a centralized MRP calculation control device 2, which first retrieves data from a data retention device 11 and

---

[1] This portion of Matoba is also cited in the rejection of claims 2, 9, 14, 16, and 18.

[2] This portion of Matoba is also cited in the rejection of claims 2-21.

Mark I. Crohn
Reg. No. 40573
109 2nd St. S., #429
Kirkland, WA 98033
(425) 828-7710

then outputs the result of the adjustment back to the data retention device. Various screen elements are then updated to reflect the changes. The display elements seem entirely conventional, including the use of bar graphs to indicate time periods in a timeline.

Dependant Claim 3

Claim 3 further recites multiple instances of the same type of component, namely a "parent action date component" and a "child action date component". Matoba does not describe multiple action date components, nor does it teach the recited association of a parent action date component and a child action date component.

The Office Action cites Matoba col. 35 lines 4-59[3]. This portion of Matoba teaches a method for sorting an order of jobs by iteratively analyzing the jobs and attempting to assign the job to alternative shops. Through this analysis, the starting date and production processing dates are determined by a very complicated method involving heaping and load accumulation graphs. This portion of Matoba does not describe the date property in the child component being set based on an offset value associated with that child component and the date property of its parent component, as is recited in claim 2.

(*See also*, the distinction of the Matoba GUI used by a human operator to adjust production planning also cited by the Office Action with regard to this claim, above at page 15).

Dependent Claim 4

Claim 4 recites an action list component that finds find no corresponding structure in Matoba. As with the other component types, Matoba also does not teach associating an instance of an action list component with an instance of an action date component, nor does Matoba recite the pass through association of the action component with the action

---

[3] This portion of Matoba is also cited in the rejection of claims 3, 5, 7, 15, 16, and 17.

Mark I. Crohn
Reg. No. 40573
109 2nd St. S., #429
Kirkland, WA 98033
(425) 828-7710

date component via the action list component. Claim 4 has been amended to clarify that the execution of a task includes performing a task method. Matoba teaches only a scheduling system and not anything that performs a task method.

The Office Action cites Matoba col. 17 lines 27-56[4]. This portion of Matoba teaches retrieving "a table 43 of order numbers with lateness for starting" in response to a human operator selecting an "Orders list menu 44." This table 43 appears to be a conventional data table returned from the date retention device and does not teach or suggest an action list component that provides the recited associations between components.

(*See also*, the distinction of the Matoba GUI used by a human operator to adjust production planning also cited by the Office Action with regard to this claim, above at page 15; the distinction of the Matoba job sorting system also cited by the Office Action with regard to this claim, above at page 16).

Dependent Claim 5

Claim 5 further recites multiple instances of the same type of component, namely a "parent action date component" and a "child action date component". Matoba does not describe multiple action date components, nor does it teach the recited association of a child action date component with an instance of the action list component. Further, Matoba does not teach the setting the date property of the child action date using the offset value associated with that child action date and the date property of the parent action date component based upon the association of the components.

(*See also*, the distinction of the Matoba GUI used by a human operator to adjust production planning also cited by the Office Action with regard to this claim, above at page 15).

---

[4] This portion of Matoba is also cited in the rejection of claims 4, 6-11, 13-14, 16, 19 and 21.

Mark I. Crohn
Reg. No. 40573
109 2nd St. S., #429
Kirkland, WA 98033
(425) 828-7710

Dependent Claim 6

Claim 6 further recites multiple instances of action list components, namely associating an instance of the action list component with another instance of the action list component. Matoba does not describe multiple action list components, nor does it teach the recited association of an action list component with another instance of an action list component. (*See also*, the distinction of the Matoba GUI used by a human operator to adjust production planning also cited by the Office Action with regard to this claim, above at page 15; the distinction of the Matoba order list table 43 also cited by the Office Action with regard to this claim, above at page 17).

Dependent Claim 7

Claim 7 recites associating multiple action list components with a parent component grouped by context. Matoba does not describe multiple action list components, nor does it teach the recited grouping of action list components by context. Further, Claim 7 recites executing the action list component based on an occurrence of the associated context. Matoba does not describe executing an action list component, nor does it describe executing an action list component based on the context it is associated with.

(*See also*, the distinction of the Matoba GUI used by a human operator to adjust production planning also cited by the Office Action with regard to this claim, above at page 15; the distinction of the Matoba order list table 43 also cited by the Office Action with regard to this claim, above at page 17; the distinction of the Matoba job sorting system also cited by the Office Action with regard to this claim, above at page 16).

Independent Claim 8

Matoba teaches only a standalone program and does not teach a "method for programming automated task lists" as is recited in Claim 8. Further, Matoba does not provide an object model to use in the programming of automated task lists comprising (1)

Mark I. Crohn
Reg. No. 40573
109 2nd St. S., #429
Kirkland, WA 98033
(425) 828-7710

an action date object having a date property that specifies a point in time and (2) an action object having an associated task method. There is no discussion in Matoba of associating or configuring objects and there is no provision made for storing associated instances of the objects as an automated task list. An "object[5]" is understood by the applicant to be a contemporary implementation of a component (discussed above) that is mostly used in the world of object-oriented programming. In particular, one source defines an object as "a variable comprising both routines and data that is treated as a discrete entity." *Microsoft Computer Dictionary* (4th Edition 1999). Matoba does not describe the recited objects according to this definition or common understanding in the art, particularly when the objects are recited as having an associated date property and an associated task method, respectively.

Matoba does not teach an association between objects in general and between an action date and action object in particular. Claim 8 has been amended to explicitly recite the industry understanding that associating an object with another object includes making the object aware of the other object such that the object can access certain of the properties, methods or events of the other object. In some computer languages this is implemented by providing a reference or a pointer to the other object that contains a memory address of that object. Associating a property or method with an object includes making the data or process associated with that property, event or method available to that object. Matoba merely groups data items together in tables, which is understood not to be an association between objects.

(*See also*, the distinction of the Matoba GUI used by a human operator to adjust production planning also cited by the Office Action with regard to this claim, above at

---

[5] Applicant notes the use of the word "object" in Matoba, but believes the that it is used in the sense of referring to a "target" of a user intervention, as no aspects understood to be part of a programming object are described.

Mark I. Cr hn
Reg. No. 40573
109 2nd St. S., #429
Kirkland, WA 98033
(425) 828-7710

page 15; the distinction of the Matoba order list table 43 also cited by the Office Action with regard to this claim, above at page 17).

Dependent Claim 9

Claim 9 recites a graphical user interface with graphic representations that allow a user to generate multiple instances of the action date object and action object as required for the programming of an automated task list. Matoba does not teach instancing action date objects or action objects based on the selection of a graphic representation of the object, nor does it teach assembling a graphical bolt representation of the automated task list using graphic representations of the instanced objects.

The Office Action cites Figures 41-46[6]. This portion of Matoba teaches methods for combining load accumulation graphs to determine starting and production processing dates. (*See also*, the distinction of the Matoba GUI used by a human operator to adjust production planning also cited by the Office Action with regard to this claim, above at page 15; the distinction of the Matoba production chain also cited by the Office Action with regard to this claim, above at page 15; the distinction of the Matoba order list table 43 also cited by the Office Action with regard to this claim, above at page 17).

Dependent Claim 10

Claim 10 recites displaying the Bolt in an alternative format as a checklist. Example B is an example of a checklist format, which is described further in the specification. Neither a Bolt nor its display is described in Matoba.

(*See also*, the distinction of the Matoba GUI used by a human operator to adjust production planning also cited by the Office Action with regard to this claim, above at page 15; the distinction of the Matoba order list table 43 also cited by the Office Action with regard to this claim, above at page 17).

---

[6] These figures of Matoba are also cited in the rejection of claims 9, 12, 14 and 16.

Mark I. Crohn
Reg. No. 40573
109 2nd St. S., #429
Kirkland, WA 98033
(425) 828-7710

Dependent Claim 11

Claim 11 recites an action list object that finds find no corresponding structure in Matoba. As with the other object types, Matoba also does not teach associating an instance of an action list object with an instance of an action date object, nor does Matoba recite the pass-through association of the action object with the action date object via the action list object.

(*See also*, the distinction of the Matoba GUI used by a human operator to adjust production planning also cited by the Office Action with regard to this claim, above at page 15; the distinction of the Matoba order list table 43 also cited by the Office Action with regard to this claim, above at page 17).

Dependent Claim 12

Claim 12 recites a graphical user interface with graphic representations that allow a user to generate multiple instances of the action date object, action list object and action object as required for the programming of an automated task list. Matoba does not teach instancing action date objects, action list objects or action objects based on the selection of a graphic representation of the object, nor does it teach assembling a graphical bolt representation of the automated task list using graphic representations of the instanced objects.

(*See also*, the distinction of the Matoba GUI used by a human operator to adjust production planning also cited by the Office Action with regard to this claim, above at page 15).

Dependent Claim 13

Claim 13 recites displaying the Bolt in an alternative format as a checklist. Example B is an example of a checklist format, which is further described in the specification. Neither a bolt nor its display is described in Matoba.

Mark I. Crohn
Reg. No. 40573
109 2nd St. S., #429
Kirkland, WA 98033
(425) 828-7710

(*See also*, the distinction of the Matoba GUI used by a human operator to adjust production planning also cited by the Office Action with regard to this claim, above at page 15; the distinction of the Matoba order list table 43 also cited by the Office Action with regard to this claim, above at page 17).

Dependent Claim 14

Claim 14 recites associating the graphic representation of the instance of the action list object with a context by attaching the instance of the action list object with the region associated with the context by way of the graphical user interface. Matoba does not describe the action list objects, let alone associating those object by way of the graphical user interface.

(*See also*, the distinction of the Matoba GUI used by a human operator to adjust production planning also cited by the Office Action with regard to this claim, above at page 15; the distinction of the Matoba production chain also cited by the Office Action with regard to this claim, above at page 15; the distinction of the Matoba order list table 43 also cited by the Office Action with regard to this claim, above at page 17).

Dependent Claim 15

Claim 15 recites storing the constituent automated task list and associating that constituent automated task list with a parent task list. An example of a behavior enabled by the aspect of the invention recited by this claim is described above at page 11, which demonstrates the reuse of previously associated objects grouped in automated task lists by adding the group to a parent automated task list. Matoba describes nothing like an automated task list nor the reuse of automated task lists as constituent automated task lists.

(*See also*, the distinction of the Matoba GUI used by a human operator to adjust production planning also cited by the Office Action with regard to this claim, above at

Mark I. Crohn
Reg. No. 40573
109 2nd St. S., #429
Kirkland, WA 98033
(425) 828-7710

page 15; the distinction of the Matoba job sorting system also cited by the Office Action with regard to this claim, above at page 16).

Dependent Claim 16

Claim 16 recites a graphical user interface with graphic representations that allow a user to generate multiple instances of the constituent bolt as required for the programming of a parent automated task list. Further, Claim 16 recites using those graphic representations to program the parent automated task list by specifying associations through the graphical user interface. Matoba does not teach instancing a constituent bolt based on the selection of a graphic representation of the constituent bolt, nor does it teach assembling a parent bolt using the graphic user interface to associate graphic representations of the parent and constituent bolts.

(*See also*, the distinction of the Matoba GUI used by a human operator to adjust production planning also cited by the Office Action with regard to this claim, above at page 15; the distinction of the Matoba production chain also cited by the Office Action with regard to this claim, above at page 15; the distinction of the Matoba order list table 43 also cited by the Office Action with regard to this claim, above at page 17; the distinction of the Matoba job sorting system also cited by the Office Action with regard to this claim, above at page 16).

Independent Claim 17

As discussed above, Matoba does not teach an automated task list and therefore could not teach executing an automated task list as recited in independent claim 17. Matoba also does not teach providing the components, setting the date properties associated with the action date components or executing pre-configured tasks associated with the action components, all as recited in claim 17.

(*See also*, the distinction of the Matoba GUI used by a human operator to adjust production planning also cited by the Office Action with regard to this claim, above at

Mark I. Crohn
Reg. No. 40573
109 2nd St. S., #429
Kirkland, WA 98033
(425) 828-7710

page 15; the distinction of the Matoba job sorting system also cited by the Office Action with regard to this claim, above at page 16).

Dependent Claim 18

Matoba does not teach an action list component, an action date component and an action list component. It does not teach the association of these components and it does not teach the execution of these components. All of these elements are recited in Claim 18 and none of them find corresponding description in Matoba.

(*See also*, the distinction of the Matoba GUI used by a human operator to adjust production planning also cited by the Office Action with regard to this claim, above at page 15; the distinction of the Matoba production chain also cited by the Office Action with regard to this claim, above at page 15).

Dependent Claim 19

Matoba does not teach an action date component or an action list component. Further, it does not teach selectively executing an associated action list component based on an analysis of pre-configured conditions.

(*See also*, the distinction of the Matoba GUI used by a human operator to adjust production planning also cited by the Office Action with regard to this claim, above at page 15; the distinction of the Matoba order list table 43 also cited by the Office Action with regard to this claim, above at page 17).

Dependent Claim 20

Claim 20 recites displaying an automated task list as a graphical bolt in the graphical user interface. Example A is an example of the bolt format, which is described in the specification. Neither an automated task list nor its display is described in Matoba.

(*See also*, the distinction of the Matoba GUI used by a human operator to adjust production planning also cited by the Office Action with regard to this claim, above at page 15).

Mark I. Crohn
Reg. No. 40573
109 2nd St. S., #429
Kirkland, WA 98033
(425) 828-7710

Dependent Claim 21

Claim 21 recites displaying the automated task list in an alternative format as a checklist. Example B is an example of a checklist format, which is described in the specification. Neither the automated task list nor its display is described in Matoba.

(*See also*, the distinction of the Matoba GUI used by a human operator to adjust production planning also cited by the Office Action with regard to this claim, above at page 15; the distinction of the Matoba order list table 43 also cited by the Office Action with regard to this claim, above at page 17).

CONCLUSION

As described above, the requirement under law that Matoba anticipate each and every element of each and every claim is clearly not met. As such, all of Claims 2-21 are allowable and Applicant respectfully requests that this application be passed to issue. The Examiner is invited to call the undersigned if he would like to discuss the Application in general or this Amendment B in particular.

//

//

Mark I. Crohn
Reg. No. 40573
109 2nd St. S., #429
Kirkland, WA 98033
(425) 828-7710

The descriptions provided in these comments are necessarily limited by space and context and are not intended to suggest that the invention is limited to the embodiments described in the included examples. Of course, the scope of the invention should be determined from the complete disclosure, the language of the claims and the inferences provided by law.

Dated this 19th day of April, 2004.

Respectfully submitted,

Mark I. Crohn
Applicant
Reg. No. 40,573
(425) 828-7710

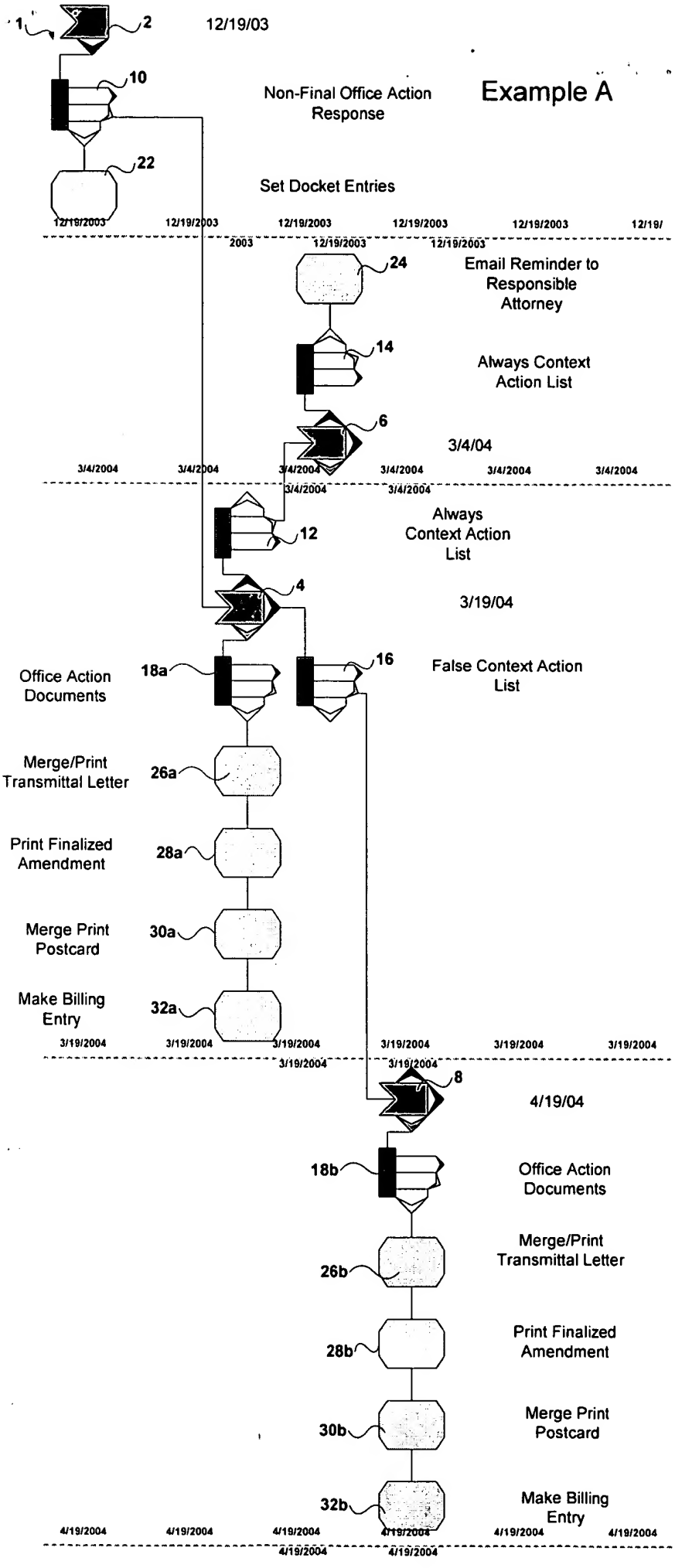
CERTIFICATE OF MAILING

I hereby certify that this communication is being deposited with the United States Postal Service, first class postage paid pursuant to 37 C.F.R. § 1.8 on the date indicated below addressed to: Box NO BOX, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450

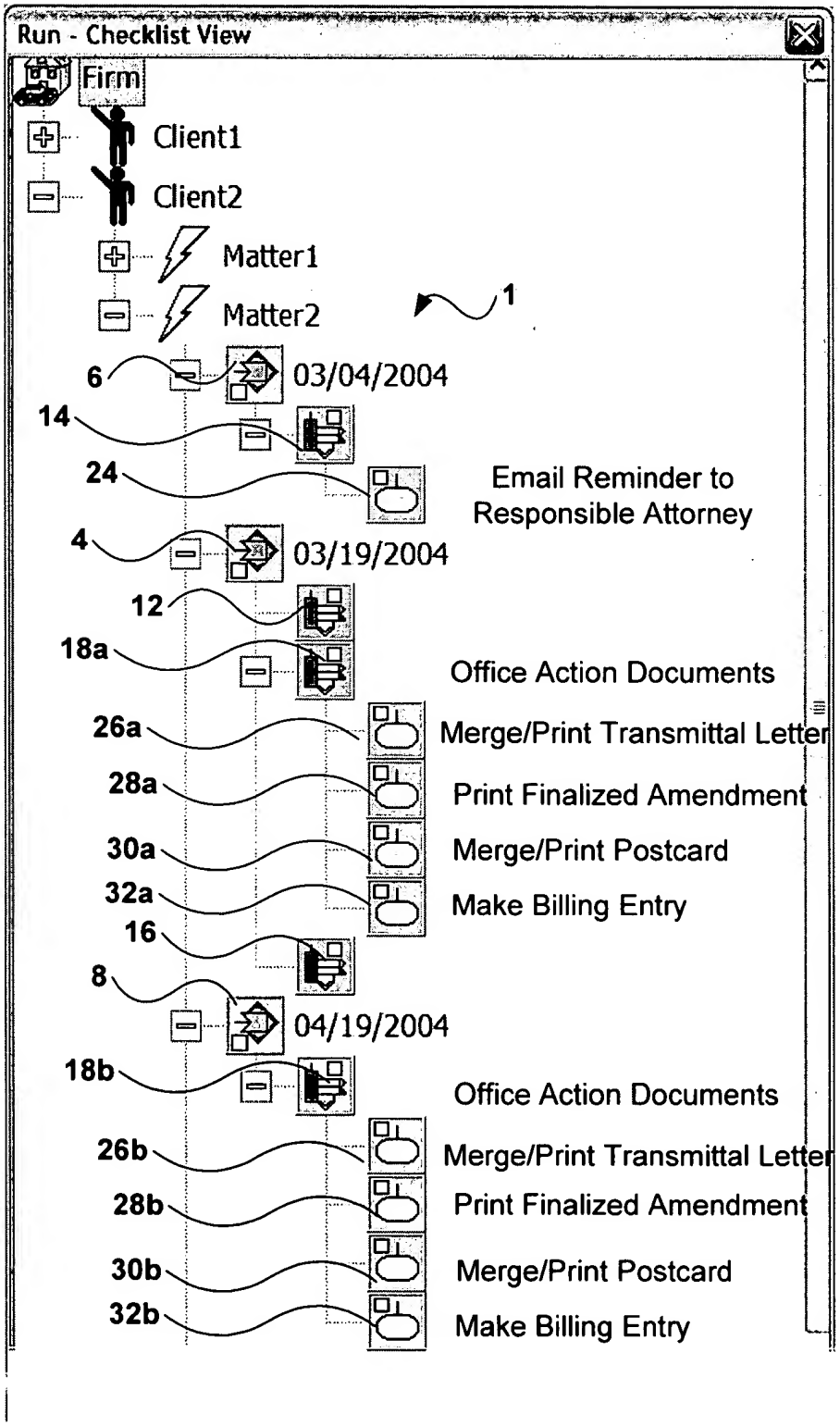
4/19/04
Date of Deposit

Mark I. Crohn

Mark I. Crohn
Reg. No. 40573
109 2nd St. S., #429
Kirkland, WA 98033
(425) 828-7710


1
2
12/19/03
10
Non-Final Office Action Response
Example A
22
Set Docket Entries
12/19/2003
12/19/2003
12/19/2003
12/19/2003
12/19/2003
12/19/2003
12/19/2003
12/19/2003
24
Email Reminder to Responsible Attorney
14
Always Context Action List
6
3/4/04
3/4/2004
3/4/2004
3/4/2004
3/4/2004
3/4/2004
3/4/2004
3/4/2004
3/4/2004
12
Always Context Action List
4
3/19/04
Office Action Documents
18a
16
False Context Action List
Merge/Print Transmittal Letter
26a
Print Finalized Amendment
28a
Merge Print Postcard
30a
Make Billing Entry
32a
3/19/2004
3/19/2004
3/19/2004
3/19/2004
3/19/2004
3/19/2004
3/19/2004
3/19/2004
8
4/19/04
18b
Office Action Documents
26b
Merge/Print Transmittal Letter
28b
Print Finalized Amendment
30b
Merge Print Postcard
32b
Make Billing Entry
4/19/2004
4/19/2004
4/19/2004
4/19/2004
4/19/2004
4/19/2004
4/19/2004
4/19/2004

Example B


Run - Checklist View
Firm
Client1
Client2
Matter1
Matter2
1
6
03/04/2004
14
24
Email Reminder to Responsible Attorney
4
03/19/2004
12
18a
Office Action Documents
26a
Merge/Print Transmittal Letter
28a
Print Finalized Amendment
30a
Merge/Print Postcard
32a
Make Billing Entry
16
8
04/19/2004
18b
Office Action Documents
26b
Merge/Print Transmittal Letter
28b
Print Finalized Amendment
30b
Merge/Print Postcard
32b
Make Billing Entry

Amended Claims Showing Additions and Deletions

2. (Amended Once) A computer-implemented system for automating a sequence of tasks, comprising:

object model components, including:

an action date component having an associated date property that specifies a point in time;

an action component having an associated task method;

means for associating an instance of the action component with an instance of the action date component; and

Deleted: and

means for executing the task method associated with the instance of the action component based on the point in time specified in the date property of the instance of the action date component to which the instance of the action component is associated; and

wherein the association of a component with another component includes making the component aware of the other component such that the component can access certain of the properties, methods or events associated with the other component.

3. (Amended Once) The system of Claim 2, wherein the instance of the action date component is a parent action date component and another instance of the action date component is a child action date component, the system further comprising:

means for associating the child action date component with the parent action date component; and

means for setting the point in time specified in the date property of the child action date component by offsetting from the point in time specified in the date property of the parent action date component by an offset value associated with the child action date component.

4. (Amended Once) The system of Claim 2, wherein the object model components include an action list component, the system further comprising:

means for associating an instance of the action list component with at least one of the instances of the action date component;

means for associating each instance of the action component with at least one instance of the action list component, thereby associating each instance of the action component with the instance of the action date component with which the instance of the action list is associated; and

means for executing the task method associated with the instance of the action component based on the point in time specified in the date property of the instance of the action date with which the instance of the action list component is associated.

8. (Amended Once) A method for programming automated task lists to be performed by a computer system, comprising:

providing an object model, including an action date object and an action object;

the action date object having a date property that specifies a point in time;

Deleted: n

the action object having an associated task method;

associating an instance of the action object with an instance of the action date object, wherein the association of a component with another component includes making the component aware of the other component such that the component can access certain of the properties, methods or events associated with the other component;

configuring the action object to perform a specific task method; and

storing associated instances of the object model objects as an automated task list.

17. (Amended Once) A method for performing an automated task list with a computer, comprising:

providing an automated task list, including:

a parent action date component having an associated parent date property;

a child action date component having an associated child date property;

the child action date component associated with the parent action date component;

an action component associated with an action date component in the bolt, the action component performing a pre-configured task method when executed;

setting the date property of action date components in the bolt by iterating the bolt from parent action date component to child action date component and adding an offset value associated with the child action date

to the value of the date property associated with the parent action date, wherein the offset value is either a positive or negative unit of time;

leaving the value of date property as previously set if the date property is marked as a hard date; and

executing each action component associated with the action date component based on the point in time specified in the date property of the action date component.